

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H03M 13/03

H03M 13/11 H03M 13/23



[12] 发明专利申请公开说明书

[21] 申请号 02148649.2

[43] 公开日 2003 年 3 月 26 日

[11] 公开号 CN 1405981A

[22] 申请日 2002. 11. 15 [21] 申请号 02148649.2

[71] 申请人 清华大学

地址 100084 北京市 100084 - 82 信箱

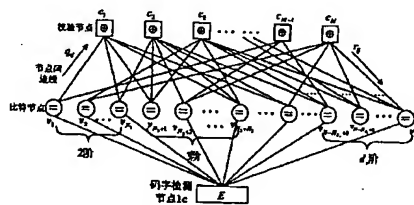
[72] 发明人 殷柳国 陆建华 吴佑寿

权利要求书 1 页 说明书 13 页 附图 3 页

[54] 发明名称 改进的非规则低密度奇偶校验码纠错译码方法

[57] 摘要

改进的非规则低密奇偶度校验码(ILDPC 码)纠错译码方法属于通信技术领域,其特征在于,它利用非规则低密度奇偶校验码比特节点的保护程度随着节点阶数的提高而提高的特性在迭代过程中使高阶比特节点的迭代在完成本阶节点的纠错时就结束,而低阶节点的迭代继续进行,以简化后续迭代的计算复杂度,节省时间;还可以通过对高阶比特节点的译码结果进行放大,给低阶节点提供更多的有用信息。从而,在高信噪比条件下可以获得比标准和积译码方法更好的纠错性能;与最小和译码方法相比,没有明显损失 ILDPC 码的纠错性能。同时,本方法还可推广到磁盘存储系统中去。



ISSN 1008-4274

1. 改进的非规则低密度奇偶校验码纠错译码方法, 含有非规则低密度奇偶校验码 (LDPC) 的和积译码方法, 它的输入为接收序列的对数似然值并在对数空间下通过利用比特节点和校验节点的约束关系进行迭代译码, 其特征在于, 它利用 LDPC 码比特节点的保护程度随着节点阶数的提高而提高的特性, 在迭代译码中, 使高阶比特节点的迭代译码在完成本阶节点的纠错时就结束, 而低阶节点的迭代译码继续进行, 以简化后续迭代译码的计算复杂度, 具体而言, 它依次含有如下步骤:

(1) 译码开始, 把接收序列输入到比特节点, 同时根据噪声大小设置各阶比特节点的迭代结束参数, 译码迭代次数置为 0;

(2) 译码器计算各阶比特节点的硬判决输出并送到码字检测节点;

(3) 检测硬判决序列是否为一个合法码字;

若是, 输出剩余各阶比特节点的译码结果, 译码结束;

若否, 则执行下一步骤;

(4) 执行下一次迭代过程: 所有各阶比特节点根据重复码的约束关系, 计算到各校验节点输出, 再通过节点间连线送到相应的校验节点作为输入; 校验节点再按照校验码的约束关系计算出反馈给各比特节点的外信息, 并把它作为比特节点下一次迭代的输入; 迭代次数加 1, 重复(2)~(4)步;

(5) 若迭代次数 $k = k_i(\sigma)$, i 为设定的高阶比特节点; 则:

(5.1) 对所有的 i 阶比特节点计算后验信息, 随后所有 i 阶比特节点把所得后验信息作硬判决, 所得结果作为译码的最后结果进行输出;

(5.2) 所有同 i 阶比特节点有联系的校验节点, 把相应的连结线简化, 进入下一次迭代运算;

(6) 迭代次数加 1, 转入下一轮迭代;

(7) 判决迭代次数是否小于最大允许值;

若是, 则回到步骤 (2);

若否, 则输出剩余各阶比特节点的译码结果;

(8) 判决是否有下一个码矢量需要译码;

若有, 则回到步骤 (1);

若无, 则结束译码过程。

改进的非规则低密度奇偶校验码纠错译码方法

技术领域

改进的非规则低密度奇偶校验码纠错译码方法属于通信信道译码技术领域，特别涉及采用前向差错控制(FEC)技术用于数据传输及存贮时的一种采用非规则低密度奇偶校验码(ILDPC码)纠正信道差错的有效而快速的译码方法。

背景技术

数据在存贮以及传输过程中经常会引发各种差错。产生这种差错的原因有随机噪声、解调过程中的同步丢失、无线传输中的多径衰落、磁性存储器中的磁道缺损等。这种突发错误一般呈非周期性出现并且持续时间长短不定。由于这些差错的存在，大大限制了特定带宽下的信息传输速率和特定面积下存储器的存储容量。特别是在无线多媒体传输系统中，由于大量的数据要在带宽有限且受到各种突发严重干扰的信道中以很高的可靠性传输，这一问题变得更加突出。

为了解决数据传输和存储中的可靠性问题，通常采用信道编码的方法。在当前已有的信道编码方法中，新近提出的ILDPC码具有最为强大的纠错能力，具有很强的应用前景。

采用ILDPC码进行差错控制的译码方法为：

1. ILDPC码的定义和参数：

ILDPC码是一种二进制分组码，这种码采用超稀疏矩阵作为校验矩阵。矩阵中每行（每列）中非零元素的个数非常稀少，且位置呈随机分布。为了便于描述，定义一行（一列）中非零元素的个数为该行（列）的重量。由于ILDPC码的校验矩阵为随机生成的矩阵，各行（列）的重量不确定，因此采用重量分布式来描述这种矩阵。同一类ILDPC码校验矩阵的列重量分布可以用分布式表示为：

$$\lambda(x) = \sum_{i=2}^{d_c} \lambda_i x^{i-1} \quad (1)$$

式中 λ_i 表示重量为 i 的列在矩阵中所占的份量， d_c 为矩阵中列重量的最大的值。同样，同一类ILDPC码校验矩阵的行重量分布采用下式描述：

$$\rho(x) = \sum_{j=2}^{d_r} \rho_j x^{j-1} \quad (2)$$

式中 ρ_j 表示重量为 j 的行在矩阵中所占的份量， d_r 为矩阵中行重量的最大值。由于ILDPC码是分组码，对于任何合法的码字 V ，与校验矩阵 H 的乘积为零，即 $H \cdot V^T = 0$ 。由该校验方程可知，校验矩阵中每列的非零元素只对应ILDPC码的同一个码元，形成了一个相当于重复码的约束。为了便于译码过程中的描述，定义这种约束关系为一个比特节点，节点的阶数即为该列的重量。而校验矩阵中每行的非零元素，将所对应的ILDPC码元映射

成一个相当于校验码的约束。同样定义这种校验关系为一个校验节点，节点的阶数即为该行的重量。矩阵中的各个非零元素，既参与了比特节点的约束关系，又参与了校验节点的约束关系，因而可以定义矩阵非零元素所对应的关系为连结这两种节点的“连结线”。在迭代译码过程中，译码器利用矩阵的行和列所对应的校验节点和比特节点的约束关系进行迭代译码。在一次迭代过程中，首先利用比特节点的约束关系进行译码，各比特节点的输入为接收序列对应的对数似然值（即各个元符号取“1”的概率除以取“0”的概率再取自然对数所得的值）以及相关校验节点在上一次迭代的输出；随后，比特节点的输出通过“连结线”送到相应的校验节点，再利用校验节点的约束关系进行译码。在这个过程中，一种节点的输出成为另外一种节点的输入，矩阵中非零元素所对应的“连结线”成为了这两种节点输入输出交换信息的“通道”。对于码长为 N 比特，列重量分布和行重量分布分别由(1)(2)两式确定的 ILDPC 码，其 i 阶比特节点的个数为：

$$N_i = N \cdot \frac{\lambda_i}{\sum_{i=2}^{d_c} \lambda_i} = N \cdot \frac{\lambda_i}{\int \lambda(x) dx}, \quad 2 \leq i \leq d_c, \quad (3)$$

同理， j 阶校验节点的个数为：

$$M_j = M \cdot \frac{\rho_j}{\sum_{j=2}^{d_v} \rho_j} = M \cdot \frac{\rho_j}{\int \rho(x) dx}, \quad 2 \leq j \leq d_c, \quad (4)$$

式中 M 为一个 ILDPC 码字中校验码元的长度。

2. ILDPC 码的译码：

ILDPC 码的译码充分应用了校验矩阵的超稀疏特性，通过比特节点和校验节点的约束关系计算并输出外信息，并相互反馈，进行迭代译码。（外信息即所有同属于一个码字的其它码元通过码字的约束关系而得到的关于某一个码元取值的信息，采用外信息交互是为了避免迭代过程中出现正反馈。）当前，ILDPC 码的译码方法主要有两种。

方法一为和积译码方法。这种方法的输入为接收序列的对数似然值，并在对数空间下通过利用比特节点和校验节点的约束关系进行迭代译码。此时，比特节点的约束关系表现为“和”的形式，即各比特节点的输出是各个输入对数似然值的和；而相应的校验节点则表现为某种“积”的形式，即各校验节点的输出是各个输入对数似然值的某种“连乘积”。由于这个特点，该方法被称之为和积译码方法。

以二元输入加性白高斯噪声信道下的信息传输系统为例，长为 $N - M$ 比特的二进制信息序列被 ILDPC 编码器编成长度为 N 比特的 ILDPC 码字。随后，该码字被调制成取值为 ± 1 的符号序列在高斯信道中传输。在接收端，接收机经过匹配滤波后得到了一串含噪声干扰的长度为 N 的实数序列 R_i^N ，随后进行信号解调。高斯信道、BPSK 调制下，第 i 个码元为 1 而经调制和传输后接收机收到信号为 R_i 的概率为：

$$P(R_i | v_i = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(R_i - 1)^2\right\}, \quad 1 \leq i \leq N \quad (5)$$

其中 σ^2 为信道噪声的标准方差。

同样，第 i 个码元为 0 而经调制和传输后接收机收到信号为 R_i 的概率为：

$$P(R_i | v_i = 0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(R_i + 1)^2\right\}, \quad 1 \leq i \leq N \quad (6)$$

由贝叶斯定理，得到：

$$P(v_i = 1 | R_i) = \frac{P(R_i | v_i = 1) \cdot P(v_i = 1)}{P(R_i)}, \quad (7)$$

$$P(v_i = 0 | R_i) = \frac{P(R_i | v_i = 0) \cdot P(v_i = 0)}{P(R_i)}. \quad (8)$$

在发送过程中，码元符号取 0 和 1 的概率相等。为了便于解调信号的输出，通常采用对数似然比的形式表示接收到的第 i 个码元取值的最大后验概率：

$$LLR(R_i) = \ln \frac{P(v_i = +1 | R_i)}{P(v_i = -1 | R_i)} \quad (9)$$

由以上各式，得：

$$\begin{aligned} LLR(R_i) &= \ln \frac{P(v_i = +1 | R_i)}{P(v_i = -1 | R_i)} = \ln \frac{P(R_i | v_i = +1)}{P(R_i | v_i = -1)} \\ &= \ln \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(R_i - 1)^2\right\}}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(R_i + 1)^2\right\}} = \frac{2}{\sigma^2} R_i \\ &= \text{sign}(R_i) \cdot \left| \frac{2}{\sigma^2} R_i \right| \end{aligned} \quad (10)$$

式中 $\text{Sign}(\bullet)$ 为符号函数。上式中，第一项符号函数表示了由接收信号得到的原发送信号取值概率的比较结果。符号函数取正值表示原码元符号为 1 的概率大于为 0 的概率；取负值则表示原码元符号为 0 的概率大于为 1 的概率。而第二项绝对值的大小则表示了该符号取 1 的概率与取 0 的概率之间的差异程度。绝对值越大，则两个概率值的差异越大。因此，(10) 式根据每个接收信号提供了两个信息，一个信息为原信号最可能取哪个值，另一个信息则表示了这种判断的可靠程度。接收机的这种解调过程充分保留了原信号的信息，被称为“软解调”，或者“软判决”，相应的软判决输出称为“软信息”。

解调器输出的软信息被送到 ILDPC 译码器进行译码。ILDPC 译码器的译码充分利用了校验矩阵的超稀疏特性，将校验矩阵的约束关系分解为行的校验码约束关系和列的重复码约束关系，通过利用这两种约束关系的相互反馈，进行迭代译码。为了便于了解在这两种

约束关系下的译码过程, 下面首先讨论重复码和校验码的译码过程。

1) 重复码的约束关系及其译码:

重复码的编码即是输入的信息符号进行 $N-1$ 次重复, 从而得到一个长为 N 的码字 V_1^N 。因此, 重复码只有两个合法码字: 全 0 码字 0_1^N 和全 1 码字 1_1^N 。经过调制、传输、解调以后, 译码器根据调制器提供的软信息进行译码。在接收到的信号序列为 R_1^N 的前提下, 根据重复码的约束关系进行译码, 得到一个采用对数似然比表示的输出序列 U_1^N 。其中, 第 i 个符号最大后验概率取值的对数似然比为:

$$\begin{aligned}
 LLR(u_i) &= \ln \frac{p(v_i = 1 | R_1^N)}{p(v_i = 0 | R_1^N)} = \ln \frac{p(v_i = 1, R_1^N)}{p(v_i = 0, R_1^N)} \\
 &= \ln \frac{\sum_{\substack{1 \leq l \leq N \\ l \neq i}} \sum_{\Lambda} p(v_1, v_2, \Lambda, v_i = 1, \Lambda, v_N, R_1^N)}{\sum_{\substack{1 \leq l \leq N \\ l \neq i}} \sum_{\Lambda} p(v_1, v_2, \Lambda, v_i = 0, \Lambda, v_N, R_1^N)} \\
 &= \ln \frac{\sum_{\substack{v_l \in V_1^N \\ l \neq i}} \sum_{\Lambda} p(v_1, v_2, \Lambda, v_i = 1, \Lambda, v_N) \cdot p(R_1^N | v_1, v_2, \Lambda, v_i = 1, \Lambda, v_N)}{\sum_{\substack{v_l \in V_1^N \\ l \neq i}} \sum_{\Lambda} p(v_1, v_2, \Lambda, v_i = 0, \Lambda, v_N) \cdot p(R_1^N | v_1, v_2, \Lambda, v_i = 0, \Lambda, v_N)} \quad (11)
 \end{aligned}$$

由于重复码只有全 0 和全 1 两个码字, 因而上式中分子分母乘积项中的第一项只有在码字分别为全 1 码字和全 0 码字时, 概率才不为 0。从而, (11) 式可以继续化简为:

$$\begin{aligned}
 LLR(u_i) &= \ln \frac{p(R_1^N | V_1^N = 1_1^N)}{p(R_1^N | V_1^N = 0_1^N)} = \ln \frac{\prod_{r=1}^N p(R_r | v_r = 1)}{\prod_{r=1}^N p(R_r | v_r = 0)} \quad (12) \\
 &= \sum_{r=1}^N LLR(v_r) = LLR(v_i) + \sum_{r \neq i} LLR(v_r)
 \end{aligned}$$

上式中结果的第一部分是码元 i 接收信号的对数似然比值, 为码元本身所具有的信息, 称为“先验信息”; 第二部分是码字中其它码元根据码字的约束关系而提供的关于码元 i 的取值信息, 称为“外信息”。由于先验信息为各个码元本身就有, 因而在译码过程中译码器只需给每个码元反馈相应的外信息。重复码的译码关系可以采用图 1 所示的一个节点图来表示。

图 1 中的节点共有 N 条连结线, 对应 N 个码元。这些连结线既可以作为输入也可以作为输出, 分别对应 N 个码元的输入和译码输出。在一次译码过程中, 接点通过 N 条连结线收到以对数似然比的形式表示的解调信息序列, 随后, 通过节点的运算, 译码结果也通过这些连结线输出 N 个码元的外信息。其中, 每条连结线的输出为其它各条连结线的输入值的累加和。在对 LDPC 码译码的描述中, 重复码的这种译码节点也被称为“比特节点”。

2) 校验码的约束关系及其译码:

以码率为 N/N 的校验码为例, 长为 $N-1$ 个比特的信息序列经过编码后得到一个码长为

N 的校验码, 码元之间的约束关系可以用如下的关系式表示:

$$v_1 \oplus v_2 \oplus \Lambda \oplus v_N = 0 \quad (13)$$

式中 \oplus 表示二进制和, 即二进制逻辑中的“异或”。校验码所得的码字 V_1^N 经过调制、传输、解调后, 得到包含该码字信息的一个软信息序列 $LLR(R_1^N)$ 。校验码译码器即根据这个软信息序列进行译码。定义二元符号 e_i 为码字中除了第 i 个码元 v_i 外其它所有码元的二进制和, 则由 (13) 式可得:

$$v_i \oplus e_i = 0 \quad (14)$$

由二进制的异或关系以及 (14) 式, 得码元 v_i 与符号 e_i 取值相同。因而, 码元 v_i 与符号 e_i 形成一个相当于重复码的关系。由上面重复码所讨论得到的结果可知, 码元 v_i 通过译码后所得的后验信息为:

$$LLR(\hat{v}_i) = LLR(v_i) + LLR(e_i) \quad (15)$$

显然, (15) 式中的第二项就是在译码过程中译码器根据整个码序列的约束关系反馈给码元 v_i 的外信息。下面, 我们以码长为 3 的校验码为例, 推导外信息的表示式。不失一般性, 我们讨论第一个码元的外信息表示式。在码长为 3 比特的情况下, 符号 e_1 取值为 1 的概率为:

$$\begin{aligned} p(e_1 = 1) &= p(v_2 = 1) \cdot p(v_3 = 0) + p(v_2 = 0) \cdot p(v_3 = 1) \\ &= p(v_2 = 1) \cdot (1 - p(v_3 = 1)) + (1 - p(v_2 = 1)) \cdot p(v_3 = 1) \\ &= p(v_2 = 1) + p(v_3 = 1) - 2p(v_2 = 1) \cdot p(v_3 = 1) \end{aligned} \quad (16)$$

从而,

$$\begin{aligned} 1 - 2p(e_1 = 1) &= 1 - 2p(v_2 = 1) - 2p(v_3 = 1) + 4p(v_2 = 1) \cdot p(v_3 = 1) \\ &= (1 - 2p(v_2 = 1)) \cdot (1 - 2p(v_3 = 1)) \end{aligned} \quad (17)$$

引入一个函数:

$$\Phi(x) = \tanh(-\frac{1}{2}x) = \frac{\exp(-\frac{1}{2}x) - \exp(\frac{1}{2}x)}{\exp(-\frac{1}{2}x) + \exp(\frac{1}{2}x)} = \frac{1 - \exp(x)}{1 + \exp(x)} \quad (18)$$

那么,

$$\Phi(LLR(e_1)) = \Phi\left(\ln \frac{p(e_1 = 1)}{p(e_1 = 0)}\right) = \frac{1 - \exp\left(\ln \frac{p(e_1 = 1)}{p(e_1 = 0)}\right)}{1 + \exp\left(\ln \frac{p(e_1 = 1)}{p(e_1 = 0)}\right)} = 1 - 2p(e_1 = 1) \quad (19)$$

由 (17)、(19) 式, 得码元 v_1 的: 外信息表示形式为:

$$LLR(e_1) = \Phi^{-1}(\Phi(LLR(v_2)) \cdot \Phi(LLR(v_3))) \quad (20)$$

(20) 式可以推广到任意一个码元, 也可以推广到码长大于 3 比特的情况。在码长为 N

比特得情况下, 码元 i 的外信息为:

$$LLR(e_i) = \Phi^{-1} \left(\prod_{\substack{1 \leq l \leq N \\ l \neq i}} \Phi(LLR(v_l)) \right) \quad (21)$$

校验码的这种译码运算关系也可以采用一个节点来表示, 如图 2 所示:

图 2 中节点具有 N 个连结线, 对应 N 个码元; 每根连结线既是输入也是输出。其中, 输入对应于输入到译码器的软信息序列, 输出则是译码器通过运算反馈给各个码元符号的外信息。在一次译码中, 每根连结线输入该码元解调后得到的软信息到节点, 随后节点通过运算, 给每个连结线一个外信息输出。注意到, 图 2 中每根连结线的输出是以其它所有连结线的输入值作为输入的运算结果。在后续 ILDPC 码的译码中, 校验码的这种译码节点被称为“校验节点”。

3) ILDPC 码的译码:

信息序列经过 ILDPC 码的编码、调制、传输后, 由接收机进行匹配滤波, 得到相应的包含 ILDPC 码字信息的接收序列 R_i^N , 随后这个序列被送到 ILDPC 码译码器进行纠错译码。在一次译码过程中, 译码器首先对接收序列进行解调, 把接收序列转化为软信息的形式; 随后, 利用 ILDPC 码的校验方程 $H \cdot V^T = 0$ 进行译码。注意到 ILDPC 码的校验矩阵为超稀疏矩阵, 每行/列的非零元素个数非常稀少。由校验方程知, 矩阵的每行 ILDPC 码的乘积, 实际上是与该行非零元素相乘的码元的二进制和。由校验码的约束方程可知, 这些码元构成了一个校验码的约束。由于校验矩阵有 M 行, 因而一共能得到 M 个校验码。通过采用校验码的译码方法, 每个校验码都可以在各自的约束关系下给各个码元一个反映该码元取值情况的外信息输出。而对于校验矩阵的每一列, 由于其元素在校验矩阵与码字的乘法中只与同一个码元相乘, 而该列的每个非零元素都对应一个校验码对该码元符号取值情况的输出。于是, 这些校验码的输出与接收到的码元软信息一起, 构成了一个重复码的约束。由于校验矩阵共有 N 列, 因而可以得到 N 个重复码, 分别与 ILDPC 码字的 N 个码元对应。ILDPC 码的译码即是通过将校验矩阵的约束关系分解为这 M 个校验码和 N 个重复码的约束关系, 通过这两种码的译码输出相互反馈为对方的输入, 进行并行迭代译码。由以上关于重复码和校验码的讨论, ILDPC 码的译码网格图可以由图 3 表示:

首先, 接收序列被译码器转化为软信息后, 译码器将所有校验节点的初始输出设为 0, 随后根据接收序列的软信息和校验节点的初始输出进行 N 个比特节点的同时译码。这些比特节点对各个码元的外信息输出, 被通过连结线送到相应的校验节点, 随后 M 个节点同时进行校验码的译码, 每个校验节点的对每个码元符号的译码输出都通过连结线反馈回相关的比特节点。在下一次迭代开始时, 每个比特节点都将自己所有的输入累加, 得到一个码元的后验信息, 随后根据这个后验信息进行硬判决译码。 N 个比特节点的硬判决译码得到一个码字的估值信息序列。如果校验矩阵与这个估值信息序列的乘积为零, 则译码器停止迭代译码并输出这个估值作为译码结果; 否则, 译码器进行下一次比特节点—校验节点的

译码迭代,直到所得估值序列为一个合法 LDPC 码字或者达到最大迭代次数为止。译码器的输出为最后一次得到的硬判决估值序列。

设 r_{ij} 为从校验节点 j 输出到比特节点 i 的外信息, q_{ij} 为从比特节点 i 到校验节点 j 的外信息,该和积译码方法的迭代过程包括如下步骤:

1)译码初始化:对于接收到的实数序列 R_i^N ,对应的 LDPC 码第 i 个码元的初始接收值被译码器解调成对数似然比的形式:

$$LLR(R_i) = \frac{2}{\sigma^2} R_i, 1 \leq i \leq N \quad (22)$$

式中 LLR 表示取值为对数似然比, σ^2 为信道噪声的标准方差。同时,初始条件下校验节点没有任何关于码字的信息,故设置校验节点 j 输出到比特节点 i 的外信息为:

$$LLR(r_{ij}) = 0 \quad (23)$$

2)若所得到的序列的硬判决结果不为一个合法的码字(其中硬判决是指根据序列各个符号的对数似然值决定各个码元符号的比特取值,对数似然值为正数则码元取符号“1”,为负数则码元取符号“0”),执行一次和积译码的迭代过程为:

a)比特节点的译码:在这种节点的约束关系下,输出与输入的关系为“和”的关系,即比特节点 i 到校验节点 j 的外信息输出为:

$$LLR(q_{ij}) = \sum_{\substack{j' \in Col[i] \\ j' \neq j}} LLR(r_{ij'}) + LLR(R_i) \quad (24)$$

式中 $Col[i]$ 表示校验矩阵 H 第 i 列非零元素的位置集合。

b)校验节点的译码:在校验节点的约束关系下,输出与输入的关系为某种“积”的关系,即校验节点 j 输出到比特节点 i 的外信息为:

$$LLR(r_{ij}) = \Phi^{-1} \left(\prod_{\substack{i' \in Row[j] \\ i' \neq i}} \Phi(LLR(q_{i'j})) \right) \quad (25)$$

式中 $Row[j]$ 表示校验矩阵 H 第 j 行非零元素的位置集合,并且

$$\Phi(x) = \tanh\left(-\frac{1}{2}x\right) \quad (26)$$

3)迭代后所得的第 i 个比特节点的译码结果为该节点所有输入的和:

$$LLR(\hat{v}_i) = \sum_{j' \in Col[i]} LLR(r_{ij'}) + LLR(R_i) \quad (27)$$

对所得到的译码结果进行如下的硬判决,然后转移到第二步。其中第 i 个码元符号的硬判决为:

$$\hat{u}_i = \begin{cases} 1 & \text{if } LLR(\hat{v}_i) > 0 \\ 0 & \text{if } LLR(\hat{v}_i) < 0 \end{cases} \quad (28)$$

4)如果需要进行下一个码字的译码,跳转到第一步;否则,结束译码。

注意到这种译码方法充分利用了比特节点和校验节点的性质,以及接收序列的所有信息,因而可以得到较好的译码性能,同时迭代过程中的收敛也比较快。但是,在码长过长(10000 比特以上)时,该方法的运算量仍然很大,难于在实际系统中应用。

另一种 ILDPC 码的译码方法为最小和译码方法。该方法的译码过程与和积译码算法相似,在比特节点的约束下输出与输入仍然为“和”的关系,但是校验节点的关系则近似简化为符号的连乘积及取最小输入绝对值的关系:

$$LLR(r_{ij}) = \min_{\substack{i' \in \text{Row}[j] \\ i' \neq i}} |LLR(q_{i'j})| \cdot \prod_{\substack{i' \in \text{Row}[j] \\ i' \neq i}} \text{sgn}(LLR(q_{i'j})) \quad (29)$$

通过采用这种近似译码方法,译码过程的计算复杂度得到较大的简化。但是,由于在校验节点的运算中采用了近似运算,丢失了较多的信息,使得纠错性能有明显的下降。在低信噪比的情况下,该方法的收敛速度很慢,与方法一相比,运算量的降低不明显,而性能却有显著的下降。

发明内容

本发明的目的在于克服现有技术的不足之处,提出了一种修正的和积译码方法。该方法注意到了 ILDPC 码对于不同阶比特节点具有不等保护成度的特点,即 ILDPC 码比特节点的保护成度随着节点阶数的提高而提高,使得在迭代译码的过程中高阶比特节点的差错往往先于低阶节点的差错被纠正过来。本发明通过利用 ILDPC 码的这种特性,在迭代译码过程中使高阶比特节点的迭代译码在完成本阶节点的纠错时就结束,而低阶节点的迭代译码继续进行,从而简化了后续迭代译码的计算复杂度。同时,在高阶比特节点迭代结束时,由于所得到的译码结果的差错水平远低于整个序列的差错水平,本发明通过对高阶比特节点的译码结果进行放大,给低阶节点提供了更多的有用信息。在高信噪比的条件下,这种方法可以获得比标准和积译码方法更好的纠错性能。从而,本发明以相对上述方法一小得多的译码复杂度和以相对方法二高得多的纠错性能,实现了 ILDPC 码的译码。

本发明的特征在于,它利用 ILDPC 码比特节点的保护程度随着节点阶数的提高而提高的特性,在迭代译码中,使高阶比特节点的迭代译码在完成本阶节点的纠错时就结束,而低阶节点的迭代译码继续进行,以简化后续迭代译码的计算复杂度,具体而言,它依次含有如下步骤:

(1)译码开始,把接收序列输入到比特节点,同时根据噪声大小设置各阶比特节点的迭代结束参数,译码迭代次数置为 0;

(2)译码器计算各阶比特节点的硬判决输出并送到码字检测节点;

(3)检测硬判决序列是否为一个合法码字;

若是，输出剩余各阶比特节点的译码结果，译码结束；

若否，则执行下一步骤；

(4)执行下一次迭代过程：所有各阶比特节点根据重复码的约束关系，计算到各校验节点输出，再通过节点间连线送到相应的校验节点作为输入；校验节点再按照校验码的约束关系计算出反馈给各比特节点的外信息，并把它作为比特节点下一次迭代的输入；迭代次数加1，重复(2)~(4)步；

(5)若迭代次数 $k = k_i(\sigma)$ ， i 为设定的高阶比特节点，则：

(5.1)对所有的 i 阶比特节点计算后验信息，随后所有 i 阶比特节点把所得后验信息作硬判决，所得结果作为译码的最后结果进行输出；

(5.2)所有同 i 阶比特节点有联系的校验节点，把相应的连结线简化，进入下一次迭代运算；

(6)迭代次数加1，转入下一轮迭代；

(7)判决迭代次数是否小于最大允许值；

若是，则回到步骤 (2)；

若否，则输出剩余各阶比特节点的译码结果；

(8)判决是否有下一个码矢量需要译码；

若有，则回到步骤 (1)；

若无，则结束译码过程。

试验证明：它达到了预期目的。

附图说明

图 1. 重复码的译码节点图。

图 2. 校验码的译码节点图。

图 3. LDPC 码的译码网格图。

图 4. 本发明所述译码方法的程序流程图。

图 5. 应用本发明纠正传输错误的通信系统框图。

具体实施方式

本发明提出的一种修正和积译码方法，如图 3 所示，假设比特节点按照阶数从低到高的顺序从左向右排列，则本发明提出的译码方法包括以下步骤：

译码开始时，接收序列输入到比特节点，译码器进行初始化，同时根据噪声的大小设置各阶比特节点的迭代结束参数，译码迭代次数置为 0。然后，译码器计算各阶比特节点的硬判决输出，送到节点码字检测节点，检测硬判决序列是否为一个合法码字。如果所得的硬判决系列为一个合法的码序列，则译码结束，输出相应的硬判决结果；否则，执行一次迭代过程：所有各阶比特节点根据重复码的约束关系计算到各校验节点的输出，通过节点间连线送到相应的校验节点作为输入；校验节点再按照校验码的约束关系计算反馈给各比特节点的外信息，并把它作为比特节点下一次迭代的输入。完成这些运算后，迭代次数加 1。

在下一次迭代开始时, 各阶比特节点再次计算硬判决输出, 随后由码字检测节点判决是否为一个合法码序列。如果是一个合法码字, 则结束译码迭代, 输出相应的硬判决序列; 否则, 执行一次迭代过程。如果当前迭代次数等于 $k_i(\sigma)$, 则所有 i 阶比特节点计算后验信息, 随后所有 i 阶比特节点将所得后验信息硬判决, 所得结果作为译码的最后结果进行输出, 这些节点不再参与后续的迭代译码。从而, 所有与 i 阶比特节点有联系的校验节点可以将相应的连结线的简化掉, 以便于下一次的迭代运算。完成这些运算以后, 迭代次数加 1, 转入下一轮迭代。显然, 在最坏的情况下, 译码器需要完成 $k_2(\sigma)$ 次迭代。

本发明所述方法的原理及算法描述如下:

1) 定义 Q_{ij} ($2 \leq i \leq d_v + 1$) 为校验矩阵第 j 行中阶数大于或等于 i 的比特节点对数似然值的乘积, 该变量主要用于在后续迭代中简化校验节点的运算。对于一个已知信噪比标准差为 σ^2 的接收序列, 选择一组迭代结束参数 $K_i(\sigma) = \{k_2(\sigma), k_3(\sigma), \Lambda, k_{d_v}(\sigma)\}$ 给各阶比特节点; 同时初始化译码方程为:

$$Q_{d_v+1,j} = 1, \quad 1 \leq j \leq M \quad (30)$$

$$LLR(R_i) = \frac{2}{\sigma^2} R_i, \quad 1 \leq i \leq N \quad (31)$$

及

$$LLR(r_{ij}) = 0 \quad (32)$$

2) 设置 $k_{d_v+1}(\sigma) = 0$ 。对于第 k 次迭代, 如果 $k > k_2(\sigma)$, 或者硬判决译码序列是一个合法码序列, 结束译码并输出硬判决序列; 否则, 执行如下迭代过程:

如果 $k_l(\sigma) < k < k_{l-1}(\sigma)$ ($l = 3, 4, \Lambda, d_v + 1$) 执行下面的步骤 a), b):

a) 比特节点的译码: 从比特节点 i 到校验节点 j 的外信息输出为:

$$LLR(q_{ij}) = \sum_{\substack{j' \in \text{Col}[i] \\ j' \neq j}} LLR(r_{ij'}) + LLR(R_i) \quad 1 \leq i \leq N - \sum_{l=1}^{d_v} N_l \quad (33)$$

b) 校验节点的译码: 校验节点 j 输出到比特节点 i 的外信息为:

$$LLR(r_{ij}) = \Phi^{-1} \left(Q_{ij} \cdot \prod_{\substack{i \leq N - \sum_{l=1}^{d_v} N_l \\ i' \in \text{Row}[j] \\ i' \neq i}} \Phi(LLR(q_{i'j})) \right) \quad (34)$$

否则, 如果 $k = k_l(\sigma)$ ($l = 2, 3, \Lambda, d_v$), 执行下面的迭代步骤:

c) 比特节点的译码: 从比特节点 i 到校验节点 j 的外信息输出改用下式计算:

$$\begin{cases} LLR(q_{ij}) = \sum_{\substack{j' \in Col[i] \\ j' \neq j}} LLR(r_{ij'}) + LLR(R_i), & 1 \leq i \leq N - \sum_{t=1}^{d_r} N_t \\ LLR(q_{ij}) = A_0 \cdot \left(\sum_{\substack{j' \in Col[i] \\ j' \neq j}} LLR(r_{ij'}) + LLR(R_i) \right), & N - \sum_{t=1}^{d_r} N_t < i \leq N - \sum_{t=l+1}^{d_r} N_t \end{cases} \quad (35)$$

式中 A_0 是一个很大的正整数, 用于放大 q_{ij} 的对数释然值, 以向低阶比特节点提供更多的有用信息。 A_0 可以在 10—1000 之间的取值, 具体取决于译码器的浮点位数。

d) 校验节点的译码: 此时校验节点 j 输出到比特节点 i 的外信息为:

$$LLR(r_{ij}) = \Phi^{-1} \left(Q_{l+1,j} \cdot \prod_{\substack{i \leq N - \sum_{t=1}^{d_r} N_t \\ i' \in Row[j] \\ i' \neq i}} \Phi(LLR(q_{i'j})) \right) \quad (36)$$

同时计算下一次迭代的 $Q_{l,j}$, 用于简化后续的迭代:

$$Q_{l,j} = Q_{l+1,j} \cdot \prod_{\substack{N - \sum_{t=1}^{d_r} N_t < i \leq N - \sum_{t=l+1}^{d_r} N_t \\ i' \in Row[j]}} \Phi(LLR(q_{ij})) \quad (37)$$

e) 1 阶比特节点的译码结果为:

$$LLR(\hat{v}_i) = \sum_{j' \in Col[i]} LLR(r_{ij'}) + LLR(R_i) \quad N - \sum_{t=1}^{d_r} N_t < i \leq N - \sum_{t=l+1}^{d_r} N_t \quad (38)$$

此时, 译码器输出相应的 1 阶比特节点所对应码元的硬判决结果为

$$\hat{u}_i = \begin{cases} 1 & \text{if } LLR(\hat{v}_i) > 0 \\ 0 & \text{if } LLR(\hat{v}_i) < 0 \end{cases} \quad N - \sum_{t=1}^{d_r} N_t < i \leq N - \sum_{t=l+1}^{d_r} N_t \quad (39)$$

注意到, 在标准和积译码算法的一次迭代运算中, 一个 i 阶比特节点要计算 i^2 次加法,

而一个 j 阶校验节点大约要执行 j^2 次浮点乘法。因此, 对于长为 N 比特的 LDPC 码的译码,

一次迭代大约需要 $\sum_{i=2}^{d_r} N_i \cdot i^3$ 次加法和大概 $\sum_{j=2}^{d_r} M_j \cdot j^3$ 次浮点乘法。而采用本发明所提出的算

法时, 在第 k 次 ($k_l(\sigma) < k \leq k_{l+1}(\sigma)$) 迭代过程中, 大约 $\sum_{t=l}^{d_r} N_t \cdot t^3$ 次加法和平均

$(\sum_{t=l}^{d_r} N_t \cdot t) / (\sum_{t=2}^{d_r} N_t \cdot t) \cdot \sum_{t=2}^{d_r} M_t \cdot t^3$ 次浮点乘法由于高阶比特节点的迭代提前结束而被省去, 从

而有效降低了译码复杂度。此外, 在本算法中, 高阶比特节点完成迭代译码时其所得的对数似然值被放大, 从而给低阶节点提供了更多的有用信息。在高信噪比的条件下, 这种方

法可以获得比标准和积译码方法更好的纠错性能。

实施例：本实施例为在清华同方 PC 机上用软件实现本发明提出的纠错译码方法，如图 4 所示，包括以下步骤：

译码开始时，译码器从步骤 4a 转到步骤 4b，接收序列输入到比特节点；做完这一步以后译码器转移到 4c，根据噪声的大小设置各阶比特节点的迭代结束参数，同时译码迭代次数置为 0，并按照 (30) (31) (32) 式进行初始化。然后，译码器转移到 4d，计算各阶比特节点的硬判决输出，在步骤 4e 进行判断。如果所得的硬判决系列为一个合法的码字，则此次译码结束，跳转到 4l，输出相应的硬判决结果；否则，转移到 4f，判断当前迭代次数是否等于 $k_i(\sigma)$ 。如果当前迭代次数不等于 $k_i(\sigma)$ ，执行一次迭代过程：各阶比特节点根据 (33) 式计算各节点的输出，通过节点间连线送到相应的校验节点作为输入；校验节点再按照 (34) 式计算反馈给各比特节点的外信息，并把它作为比特节点下一次迭代的输入。完成这些运算后，转移到步骤 4j。如果当前迭代次数等于 $k_i(\sigma)$ ，则转移到 4h。此时比特节点按照 (35) 式计算外信息，随后所有 i 阶比特节点结束迭代并将所得译码结果硬判决输出；与 i 阶比特节点有联系的校验节点按照 (36) 式计算外信息。随后在步骤 4i，译码器根据 (37) 式简化校验节点，以便于下一次的迭代运算。完成这些步骤以后，译码器转移到 4j，迭代次数加 1，并在步骤 4k 判断迭代次数是否小于允许值。如果是，则跳转到 4d，转入下一轮迭代；否则，转移到步骤 4l，输出剩余的各阶比特节点的译码结果。完成步骤 4l 的操作以后，译码器转移到步骤 4m，判断译码过程是否结束：如果是，则下一步转移到步骤 4n，结束译码过程；否则下一步跳回到步骤 4b，重新开始下一个码矢量的译码。

作为一个例子，表 1 和表 2 分别列出了一个 LDPC 码在 BIAWGN 信道下采用和积译码算法以及本发明的算法所得到的译码性能以及相应的计算复杂度。该 LDPC 码的主要参数为：码长等于 10000 比特，列重量分布式为 $\lambda(x) = 0.23882x + 0.29515x^2 + 0.03261x^3 + 0.43342x^{10}$ ，行重量分布式为 $\rho(x) = 0.43011x^6 + 0.56989x^7$ 。由表 1 可见，在低信噪比的条件下，两者的纠错性能相差不多；在高信噪比的条件下，本发明的算法所得的纠错性能要比标准和积译码算法好一些。另外，从表 2 可知，本发明的方法是译码复杂度明显降低。其中加法大约降低 45%—70%，浮点乘法减少了 25%—40% 左右。

表 1. 两种译码算法在 BIAWGN 信道下的性能

E_b/N_0 (dB)	0.72	0.82	0.91	1.01
和积译码算法	2.725e-3	5.858e-4	1.364e-4	5.94e-5
本发明的算法	3.241e-3	6.614e-4	1.474e-4	5.38e-5

表 2. 两种译码算法的平均译码复杂度

算法 E_b/N_0 (dB)	和积译码算法		本发明的算法	
	加法数	乘法数	加法数	乘法数
0.72	3.29×10^8	3.34×10^8	0.94×10^8	2.06×10^8
0.82	2.02×10^8	2.05×10^8	0.83×10^8	1.40×10^8
0.91	1.39×10^8	1.41×10^8	0.76×10^8	1.06×10^8
1.01	1.08×10^8	1.10×10^8	0.58×10^8	0.82×10^8

可见, 采用本方法能够以很低的译码复杂度得到很好的纠错性能, 大大提高了 ILDPC 码的实用性。

参照图 5, 采用本发明的方法纠正传输错误的通信系统包括一个产生数字信息流的信源 51, ILDPC 编码器 53, 传输信道 55, 以及如图 3 所示的纠正传输错误译码器 57。在本例中, 信源 51 产生的携带信息的数据符号流 52 被送往 ILDPC 码编码器 53, ILDPC 码编码器 53 对信息进行信道编码。编码后的 ILDPC 码流 54 在传输信道 55 传输过程中受到干扰而产生差错, 包含传输差错的码流 56 被纠正传输错误的 ILDPC 码译码器 57 所接收。经过 ILDPC 码译码器 57 采用本发明的方法完成纠错译码, 输出的码流 58 为正确的数字信息流。

应当指出, 本发明方法的应用还可以推广到磁存储系统中去。

本发明的效果是, 通过利用 ILDPC 码的不同阶比特节点的不等差错保护特性, 使得高阶比特节点的迭代比低阶比特节点的迭代先结束, 从而在没有明显损失译码性能的前提下将译码复杂度明显降低。与已有的方法一相比, 本方法显著降低了计算复杂度; 与已有的方法二相比, 本方法没有明显损失 ILDPC 码的纠错性能。此外, 本方法还通过对高阶比特节点对数似然值进行适度放大, 使得低阶节点获得了更多的有用信息。在高信噪比的条件下, 可以获得比方法一更好的纠错性能。因此, 对于 ILDPC 码的译码, 本方法明显优于其它的方法。

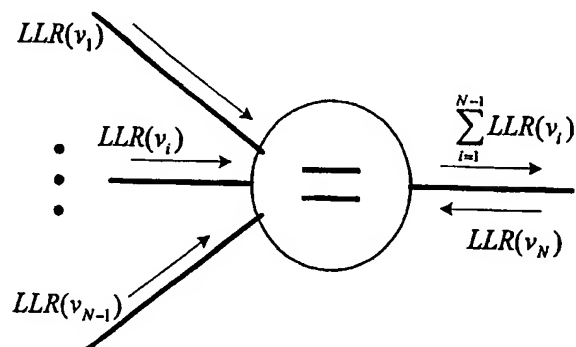


图 1

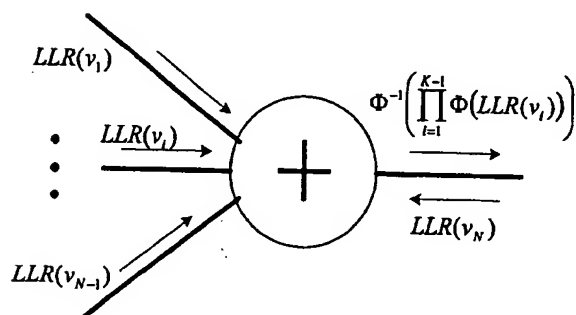


图 2

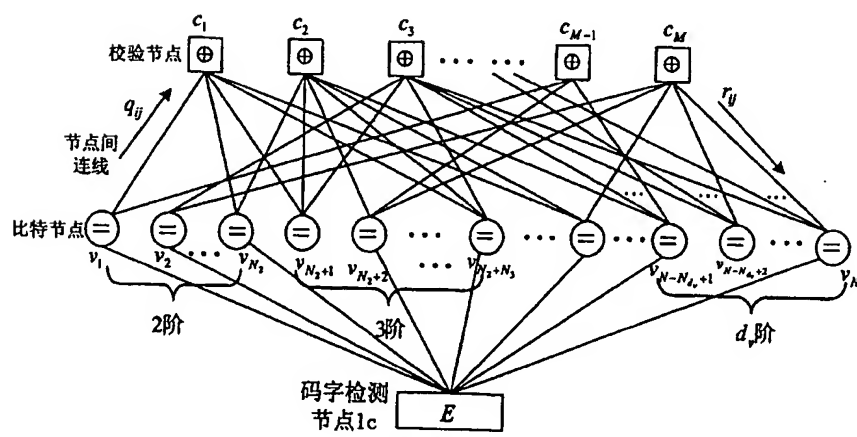


图 3

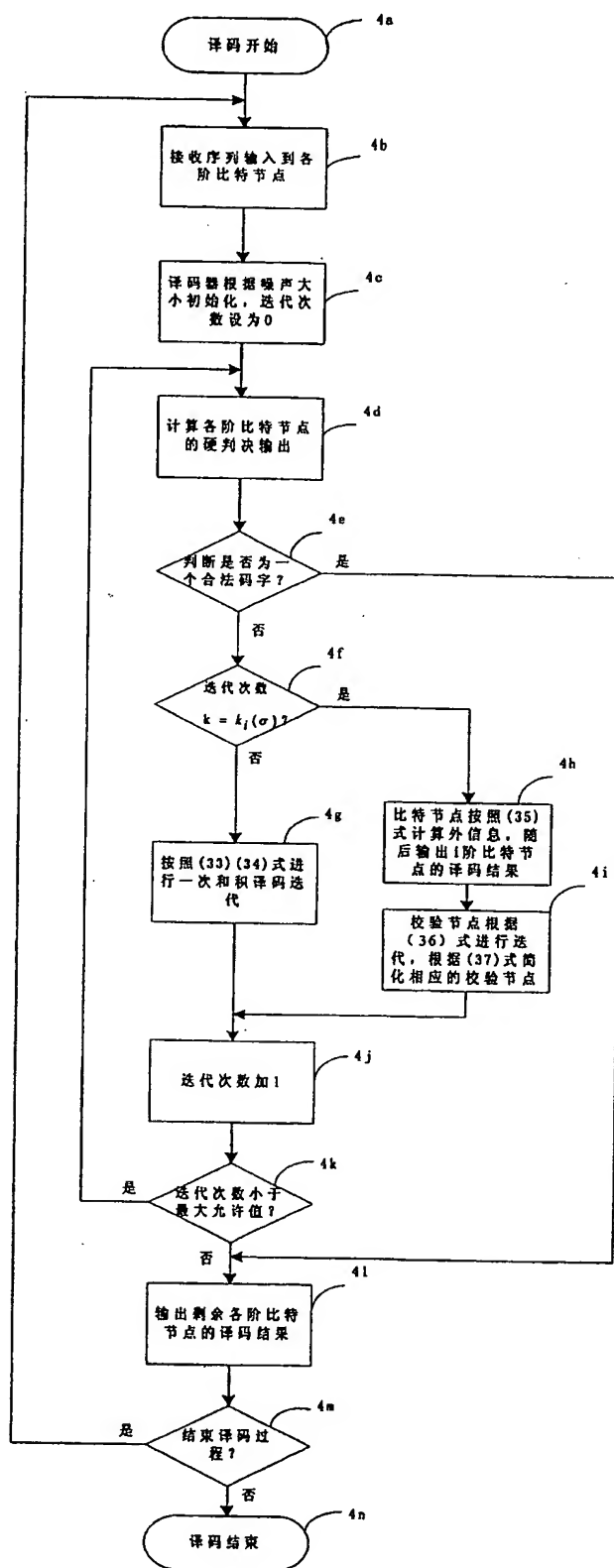


图4

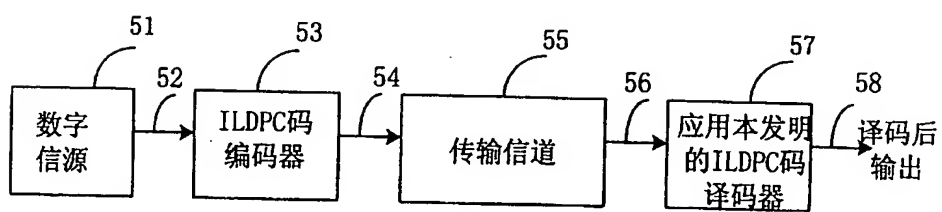


图 5

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.